Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application. Claims 1, 3, 4, 7, 13, 17, 18, 20, and 21 have been amended and new Claim 23 has been added.

Listing of Claims:

1. (Currently Amended): A method, comprising:

determining whether each of ~~a bootstrap processor from~~ a plurality of [operable] processors in a fault tolerant multiprocessor system ~~irrespective of an initialization time of a particular operable processor~~ is operable;

~~asserting a first signal;~~

~~asserting a second signal; and~~

wherein said determining further comprises determining whether the processor is asserting a stall signal on a system bus; and

~~ensuring that both the first signal and the second signal are asserted prior to~~ selecting a bootstrap processor from among [allowing the plurality of] the operable processors ~~to enter a bootstrap processor arbitration process~~ irrespective of an initialization time of a particular operable processor.

2. (Canceled)

3. (Currently Amended): The method of claim 1, [wherein] further comprising:

~~the first signal indicates that a particular processor has successfully completed an initialization sequence, and the second signal indicates that~~ determining whether all of the operable processors ~~in the fault tolerant multiprocessor system~~ are ready to enter [the] a bootstrap processor arbitration process.

4. (Currently Amended): The method of claim [1]3, wherein determining whether all of the operable processors are ready further comprises: ~~the second signal is communicated across a system bus to each processor~~.

determining whether each processor has de-asserted the stall signal on the system bus; and

determining whether a request queue indicates a pending transaction for the system bus.

5. (Previously Presented): The method of claim 1, wherein asserting is selected from one in a group consisting of driving a signal line to a logical 0, driving the signal line to a logical 1, toggling the signal line from a logical 1 to a logical 0, and driving the system bus to a logic state on a first clock cycle and releasing the system bus on a second clock cycle.

6. (Previously Presented): The method of claim 1, wherein the fault tolerant multiprocessor system comprises a multiprocessor system which continues to be operable irrespective of a fault occurring in any particular processor.

7. (Currently Amended): The method of claim 1, wherein determining whether each processor is [the] operable further comprises determining whether each processor [comprises a processor which] has successfully completed an initialization and testing sequence.

8-12. (Canceled)

13. (Currently Amended): A computing system, comprising:

a plurality of ~~operable~~ processors;

a system bus coupled to each of the processors;

an arbitration protocol to determine a bootstrap processor from the plurality of ~~operable~~ processors ~~in a fault tolerant multiprocessor system~~ irrespective of an initialization time of a particular ~~operable~~ processor; and

logic to ensure that a) a stall signal has been ~~both a first signal and a second signal are~~ de-asserted on the system bus by each processor and that b) at least one processor is operable prior to allowing the [plurality of]operable processors to enter a bootstrap processor arbitration process.

14. (Original): The computing system of claim 13, wherein the arbitration protocol comprises micro code instructions.

15. (Original): The computing system of claim 13, wherein the arbitration protocol comprises logic circuitry located in a processor.

16. (Original): The computing system of claim 13, wherein the arbitration protocol conducts the bootstrap processor arbitration process across the system bus.

17. (Currently Amended): The computing system of claim 13, wherein each of the [operable] processors has a bus controller, the bus controller to stall transactional activity on the system bus until all operable processors are ready for the bootstrap processor determination [has been made].

18. (Currently Amended): An apparatus, comprising:

a computer readable media; and

instructions embedded on the computer readable media, the instructions when executed by a machine, cause the machine to perform operations comprising:

determining whether ~~a bootstrap processor from~~ each of a plurality of ~~operable~~ processors in a fault tolerant multiprocessor system ~~irrespective of an initialization time of a particular operable processor~~ is contributing to assertion of a signal on a system bus; and

~~asserting a first signal;~~

~~asserting a second signal; and~~

allowing one or more selected ones of the plurality of processors to enter a bootstrap processor arbitration process responsive to [ensuring that] both 1) the [first] signal has been de-asserted on the system bus by each of the plurality of processors and 2) the selected processors are operable. ~~the second signal are asserted prior to allowing the plurality of operable processors to enter a bootstrap processor arbitration process.~~

19. (Canceled)

20. (Currently Amended): The apparatus of claim 18, wherein de-assertion of the [first] signal by a particular processor indicates that the [a] particular processor has completed an initialization sequence~~, and the second signal indicates that all of the operable processors in the fault tolerant multiprocessor system are ready to enter the bootstrap processor arbitration process~~.

21. (Currently Amended): The apparatus of claim 18, wherein ~~the first signal indicates that a particular processor has been assigned an arbitration identification number~~ the instructions when executed by a machine, further cause the machine to determine that a particular one of the processors is operable if the particular processor has 1) de-asserted the signal and 2) indicated successful completion of an initialization sequence.

22. (Previously Presented): The apparatus of claim 18 wherein the instructions comprise micro code.

23. (New) The apparatus of claim 21, wherein the instructions that cause the machine to determine if a particular processor has indicated successful completion of an initialization sequencer further comprise instructions that cause the machine to determine whether a transaction request for the system bus has been pended.